DISPLAY DE CRISTAL LIQUIDO (LIQUID CRYSTAL DISPLAY – LCD)

Este periférico é especificamente construído para ser usado com microcontroladores, o que significa que este não pode ser ativado por circuitos integrados padrões. Este é usado para mostrar diferentes mensagens em um display de cristal liquido miniatura. O modelo descrito aqui é o mais usado pelo baixo preço e grandes capacidades.

Este LCD é baseado no microcontrolador HD44780 (*Hitachi*) e pode mostrar mensagens em duas linhas com 16 caracteres cada. Este pode mostrar todas as letras do alfabeto, letras gregas, marcas de pontuação, símbolos matemáticos etc. Este também pode mostrar símbolos feitos pelo usuário. Outras características úteis incluem deslocamento de mensagem automaticamente (esquerda e direita), aparência do cursor, luz de fundo com LED etc.

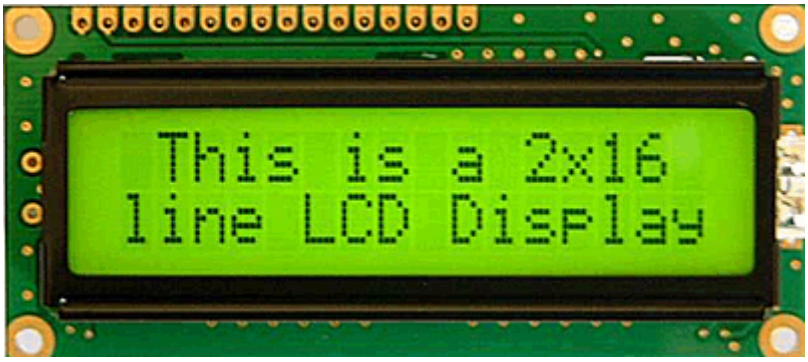

Figura 1. LCD 2x16

Pinos do Display LCD

Ao longo de um lado do circuito impresso do LCD existem pinos que habilitam estes para ser conectados a um microcontrolador. Há um total de 14 pinos marcados com números (16 se há luz de fundo – backlight). Suas funções são descritas na seguinte tabela:

Função	Pino	Nome	Estado lógico	Descrição
Terra	1	Vss	-	0 V
Alimentação	2	Vdd	-	+5 V
Contraste	3	Vee	-	0 a Vdd
Controle de Operação	4	RS	0	D0 a D7 são interpretados como comandos
			1	D0 a D7 são interpretados como dados
	5	R/W	0	Escrever dados (desde o controlador para o
			1	Ler dados (desde o LCD para o controlador)
	6	E	0	Acesso ao LCD desabilitado
			1	Operação normal
			Desde 1 para 0	Dados/comandos são transferidos para o LCD
Dados / Comandos	7	D0	0/1	Bit 0 - LSB
	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
	14	D7	0/1	Bit 7 - MSB

Tela do LCD

Uma tela do LCD pode mostrar duas linhas com 16 caracteres cada. Cada caractere consiste de uma matriz de pontos de 5x8 ou 5x11. Aqui será coberto o display com caracteres 5x8 os quais são os mais comuns.

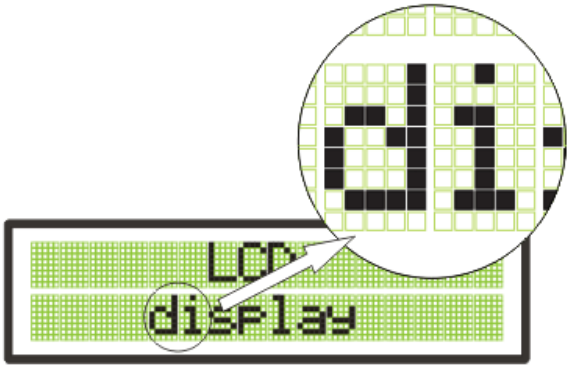

Figura 2. LCD com matriz de pontos 5x8

O contraste do display depende da tensão de alimentação e se as mensagens são mostradas em uma ou duas linhas. Por esta razão, uma variação de voltagem de 0 a Vdd é aplicado ao pino marcado como Vee. Um potenciômetro é usado para este propósito. Alguns dos LCDs têm embutidos backlights (LEDs azul ou verde). Quando se usa este backlight se coloca um resistor limitador de corrente em série a um dos pinos que alimenta o backlight como com os leds.

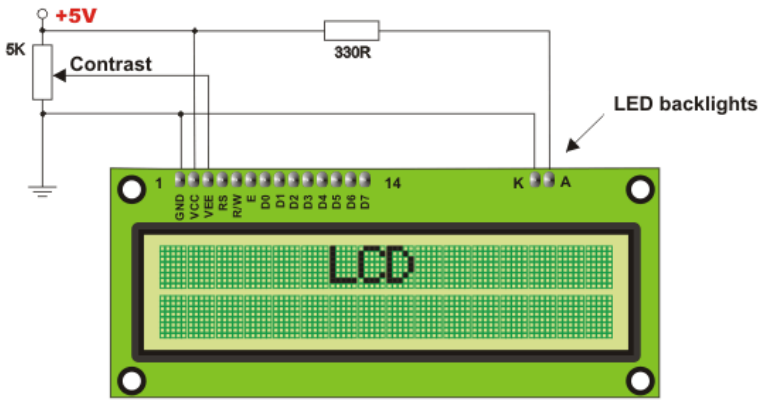

Figura 3. Circuito de conexão do backlight do LCD

Se não se mostram caracteres ou se todos eles estão quase apagados quando o display está ligado, a primeira coisa a ser feita é checar o potenciômetro de contraste para ajustar este apropriadamente.

Memória do LCD

O LCD contém três blocos de memória:

- Display Data RAM – DDRAM
- Character Generator RAM – CGRAM
- Character Generator ROM – CGROM

Memória DDRAM

A memória DDRAM é usada para armazenar caracteres a serem mostrados. O tamanho desta memória é capaz de armazenar 80 caracteres (quarenta caracteres para cada linha). Algumas locações de memória são diretamente conectadas aos caracteres no display.

Qualquer trabalho por mais simples que seja será suficiente para configurar o display para incrementar automaticamente o endereço (deslocar para direita) e setar o endereço de inicio para que a mensagem seja mostrada (por exemplo 00 hex).

Depois, todos os caracteres enviados através das linhas D0 – D7 serão mostrados no formato da mensagem que estejamos usando desde esquerda à direita. Neste caso, se começa a mostrar desde o primeiro campo da primeira linha porque é o endereço inicial 00 hex. ***Se mais de 16 caracteres são enviados, então todos eles serão memorizados, mas somente os primeiros 16 caracteres estarão visíveis***. Para mostrar o resto deles, o comando de deslocamento será usado. Virtualmente, todo o que se vê no LCD é uma janela a qual se desloca de esquerda – direita sobre as locações de memória contendo diferentes caracteres. Na verdade, isto é como o efeito da mensagem se deslocando sobre a tela que tem sido criada.

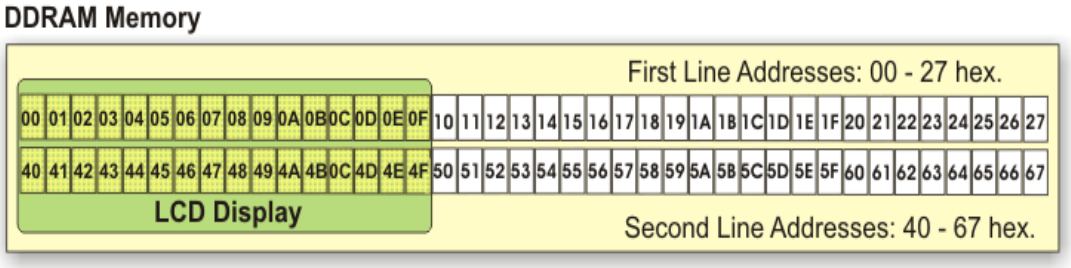

Figura 4. Memória DDRAM

Se o cursor está ligado, este aparece na locação de endereço atual. Em outras palavras, quando um caractere aparece na posição do cursor este automaticamente se moverá para a próxima posição de memória.

Este é um tipo de memória RAM tal que os dados podem ser gravados e lidos a partir dele, mas seu conteúdo é irremediavelmente perdido quando a energia é desligada.

Memória CGROM

A memória CGROM contém um mapa de caracteres padrão com todos os caracteres que podem ser exibidos na tela. Cada caractere é atribuído a uma posição de memória:

Figura 5. Memória CGROM com os caracteres ASCII

Os endereços das posições de memória CGROM correspondem aos caracteres ASCII. Se o programa que está sendo executado encontra um comando *'envie caractere P para a porta'*, então o valor binário 0101 0000 aparece na porta. Esse valor é o equivalente em ASCII para o caractere P. Em seguida, é escrito para o LCD, o que resulta em exibir o símbolo da posição da CGROM 0101 0000. Em outras palavras, o caractere 'P' é apresentado. Isso se aplica a todas as letras do alfabeto (maiúsculas e minúsculas), mas não a números. Como visto no mapa anterior, os endereços de todos os dígitos são

empurrados para frente por 48 em relação aos seus valores (o endereço do dígito 0 é 48, endereço do dígito 1 é 49, o endereço do dígito 2 é 50, etc.). Assim, a fim de exibir os dígitos corretamente, é necessário somar o número decimal 48 para cada um deles antes de serem enviados para o LCD.

O que é ASCII? De sua criação até hoje, os computadores podem reconhecer apenas números, mas não as letras. Isso significa que todas as trocas de dados de um computador para um periférico têm um formato binário, embora o mesmo seja reconhecido pelo homem como letras (o teclado é um excelente exemplo). Em outras palavras, cada caractere corresponde a uma única combinação de zeros e uns. ASCII é a codificação de caracteres baseado no alfabeto Inglês. Código ASCII especifica uma correspondência entre os símbolos de caracteres padrão e seus equivalentes numéricos.

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
0	0	NUL	16	10	DLE	32	20	(space)	48	30	0
1	1	SOH	17	11	DC1	33	21	!	49	31	1
2	2	STX	18	12	DC2	34	22	"	50	32	2
3	3	ETX	19	13	DC3	35	23	#	51	33	3
4	4	EOT	20	14	DC4	36	24	$	52	34	4
5	5	ENQ	21	15	NAK	37	25	%	53	35	5
6	6	ACK	22	16	SYN	38	26	&	54	36	6
7	7	BEL	23	17	ETB	39	27	'	55	37	7
8	8	BS	24	18	CAN	40	28	(	56	38	8
9	9	TAB	25	19	EM	41	29	)	57	39	9
10	A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	C	FF	28	1C	FS	44	2C	,	60	3C	<
13	D	CR	29	1D	GS	45	2D	-	61	3D	=
14	E	SO	30	1E	RS	46	2E	.	62	3E	>
15	F	SI	31	1F	US	47	2F	/	63	3F	?

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	|
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	

Figura 6. Tabela com o código ASCII

Memória CGRAM

Além de caracteres padrão, o display LCD também pode exibir símbolos definidos pelo próprio usuário. Pode ser qualquer símbolo no tamanho de 5x8 pixels. A memória RAM de 64 bytes de tamanho chamada CGRAM permite isso.

Os registradores da memória são de oito bits, mas apenas os cinco bits mais baixos são usados. Um 1 lógico em uma posição de cada registro representa um ponto, enquanto que 8 posições agrupadas representar um caractere. Isto é mais bem ilustrado na figura 7.

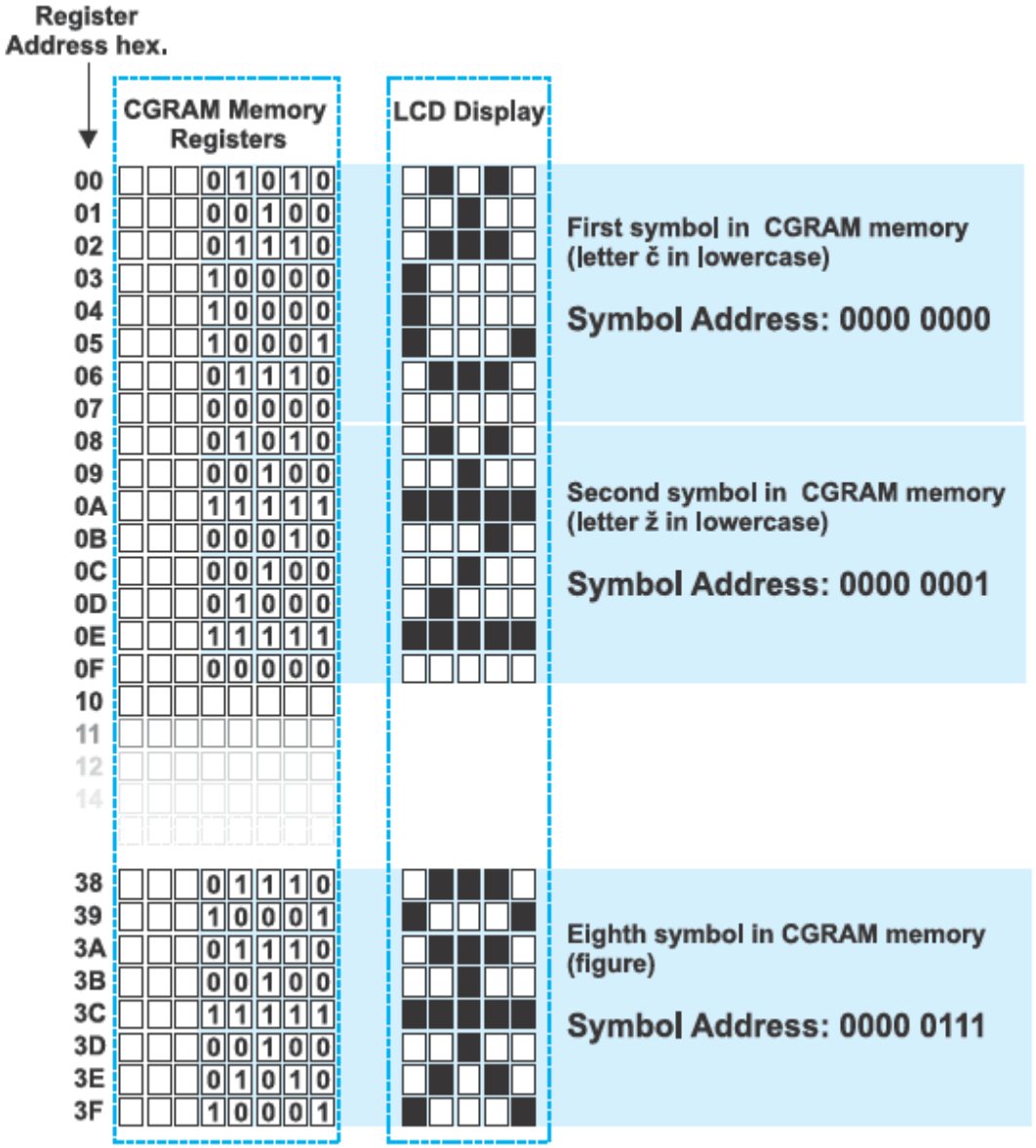

Figura 7. Memória CGRAM

Os símbolos geralmente são definidos no início do programa pela simples escrita de zeros e uns nos registradores da memória CGRAM de modo que formam as formas desejadas. A fim de exibi-los, basta especificar o seu endereço. Preste atenção à primeira coluna no mapa de caracteres da CGRAM. Ela não contém endereços de memória RAM, mas símbolos que estão sendo discutidos aqui. Neste exemplo, “mostrar 0” significa – mostrar ' č ', “mostrar 1” significa – mostrar ' ž ' etc.

Comandos básicos do LCD

Todo dado transferido para um LCD através das saídas D0 – D7 será interpretado como um comando ou um dado, o qual depende do estado lógico do pino RS.

- RS = 1 – Os bits D0 – D7 são endereços dos caracteres a serem mostrados. O processador do LCD endereça um caractere desde o mapa de caracteres e mostra este. O endereço DDRAM especifica a posição na qual o caractere está para ser mostrado. Este endereço é definido antes da transferência do caractere ou o endereço do caractere anteriormente transferido é automaticamente incrementado.
- RS = 0 – Os bits D0 – D7 são comandos para setar os modos do display.

A continuação uma lista de comandos reconhecidos pelo LCD:

COMMAND	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	EXECUTION TIME(8BITS/4BITS)
Clear display	0	0	0	0	0	0	0	0	0	1	1.64ms/4,9ms
Cursor home	0	0	0	0	0	0	0	0	1	x	1.64ms/4,8ms
Entry mode set	0	0	0	0	0	0	0	1	**I/D**	**S**	40 µs/120µs
Display on/off control	0	0	0	0	0	0	1	**D**	**U**	**B**	40 µs/120 µs
Cursor / Display Shift	0	0	0	0	0	1	**D/C**	**R/L**	x	x	40 µs/120 µs
Function set	0	0	0	0	1	**DL**	**N**	**F**	x	x	40 µs/120 µs
Set CGRAM address	0	0	0	1	CGRAM address						40 µs/120 µs
Set DDRAM address	0	0	1	DDRAM address							40 µs/120 µs
Read "BUSY" flag (BF)	0	1	**BF**	DDRAM address							1.64ms/4,9ms
Write to CGRAM or DDRAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	40 µs/120 µs
Read from CGRAM or DDRAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	40 µs/120 µs

Se **I/D** = 1 (Incrementa 1), **I/D** = 0 (Decrementa 1)

Se **R/L** = 1 (Desloca a direita), **R/L** = 0 (Desloca a esquerda)

Se **S** = 1 (Desloca display ON), **S** = 0 (Desloca display OFF)

Se **DL** = 1 (Interface de 8 bits), **DL** = 0 (Interface de 4 bits)

Se **D** =1 (Display ON), **D** = 0 (Display OFF)

Se **N** = 1 (Display em duas linhas), **N** = 0 (Display em uma linha)

Se **U** = 1 (Cursor ON), **U** = 0 (Cursor OFF)

Se **F** = 1 (Formato do caractere 5x10 pontos), **F** = 0 (Formato do caractere 5x7 pontos)

Se **B** = 1 (Cursor pisca), **B** = 0 (Cursor não pisca)

Se **D/C** = 1 (desloca display), **D/C** = 0 (desloca cursor)

Que é o flag ocupado (Busy Flag)?

Comparado ao microcontrolador o LCD é um componente extremamente lento. Por esta razão, foi necessário fornecer um sinal o qual poderia, na execução de um comando, indicar que o display está pronto para o próximo dado. Este sinal, chamado de busy flag, pode ser lido desde a linha D7. O display está pronto para receber novos dados quando a tensão nesta linha é 0V (BF = 0).

Conectando o LCD

Dependendo de quantas linhas são usadas para conectar um LCD a um microcontrolador se tem os modos de 8 bits e de 4 bits. O modo apropriado é selecionado no começo da operação no processo chamado "inicialização". O modo 8 bits usa as saídas D0 – D7 para transferir dados. O principal propósito do modo 4 bits do LCD é economizar os valiosos pinos I/O do microcontrolador. Somente o nibble superior (4 bits superiores D4 – D7) são usados para a comunicação, enquanto que o nibble inferior (D0 – D3) podem ser deixados desconectados. Cada pedaço de dados é enviado ao LCD em dois passos – Os quatro bits superiores são enviados primeiro (D4 – D7), depois os quatro bits menos significativos (D0 – D3). A inicialização habilita o LCD para enlaçar e interpretar corretamente os bits recebidos.

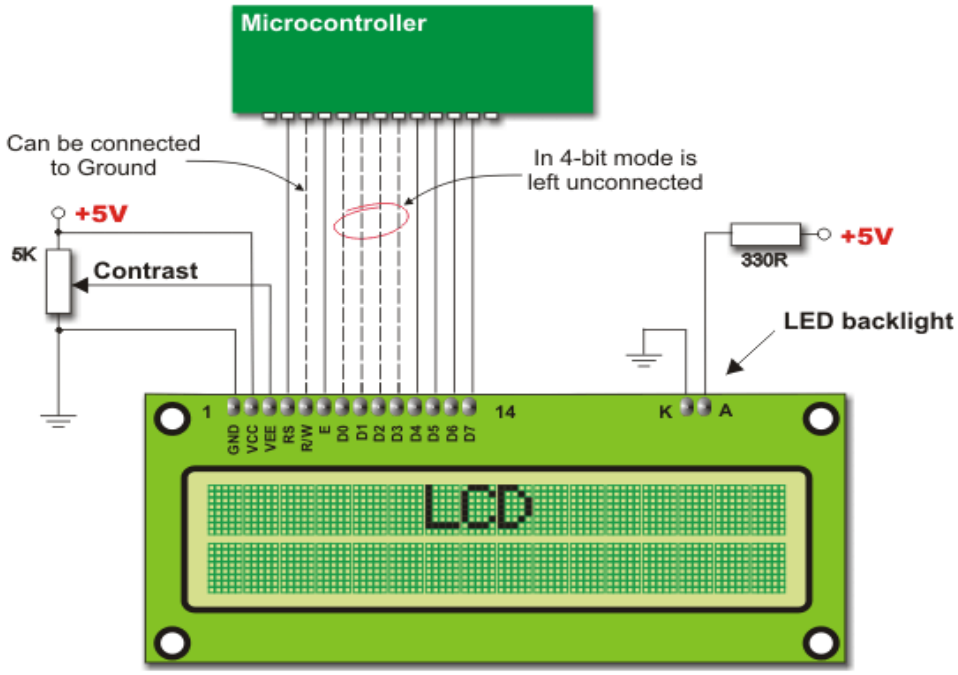

Figura 8. Conexões do LCD com o microcontrolador (modo 4 bits – linha cheia, modo 8 bits – linha cheia + linha pontilhada).

O dado é raramente lido desde o LCD (este é transferido do microcontrolador para o LCD), por isso muitas vezes é possível economizar um pino I/O extra através de uma

simples ligação do pino R / W para terra. Tal economia tem o seu preço. As mensagens serão exibidas normalmente, mas não será possível ler o busy flag já que não é possível ler a tela também. Felizmente, existe uma solução simples. Após o envio de um caractere ou um comando é importante dar o suficiente tempo ao LCD para fazer o seu trabalho. Devido ao fato de que a execução de um comando pode durar aproximadamente 1.64 mS, será suficiente esperar cerca de 2ms pelo LCD.

Inicialização do LCD

O LCD é automaticamente limpado (clear) quando ligado. Ela dura cerca de 15 ms. Depois disso, ele está pronto para operação. O modo de operação é definido por padrão (default), o que significa que:

1) Display é limpado

2) Modo:

DL = 1 – Comunicação através da interface de 8 bits

N = 0 – Mensagens são mostradas em uma linha

F = Fonte do caractere 5 x 8 pontos

3) Display/Cursor ligado/desligado (on/off)

D = 0 – Display off

U=0 – Cursor off

B = 0 Cursor piscando (blink) off

4) Caracteres de Entrada

ID = 1 – Endereços apresentados são automaticamente incrementados por 1.

S = 0 – Deslocamento do display off

O reset automático ocorre principalmente sem problemas. Principalmente, mas não sempre!! Se por alguma razão a tensão de alimentação não alcança seu valor completo dentro de 10 ms, o display começará a funcionar de forma imprevisível. Se a unidade de alimentação de tensão proporciona a tensão necessária, então, um funcionamento completamente seguro é previsto e o processo de inicialização é aplicado. Inicialização, entre outras coisas, provoca um novo reset, permitindo ao display funcionar normalmente.

Existem dois algoritmos de inicialização. A escolha de um deles depende se o microcontrolador vai ser conectado no modo de interface de 4 ou 8 bits. Em ambos os casos, tudo o que resta a fazer após a inicialização é especificar os comandos básicos e, é claro - para exibir as mensagens.

A figura 9 mostra o algoritmo de inicialização no modo de 8 bits.

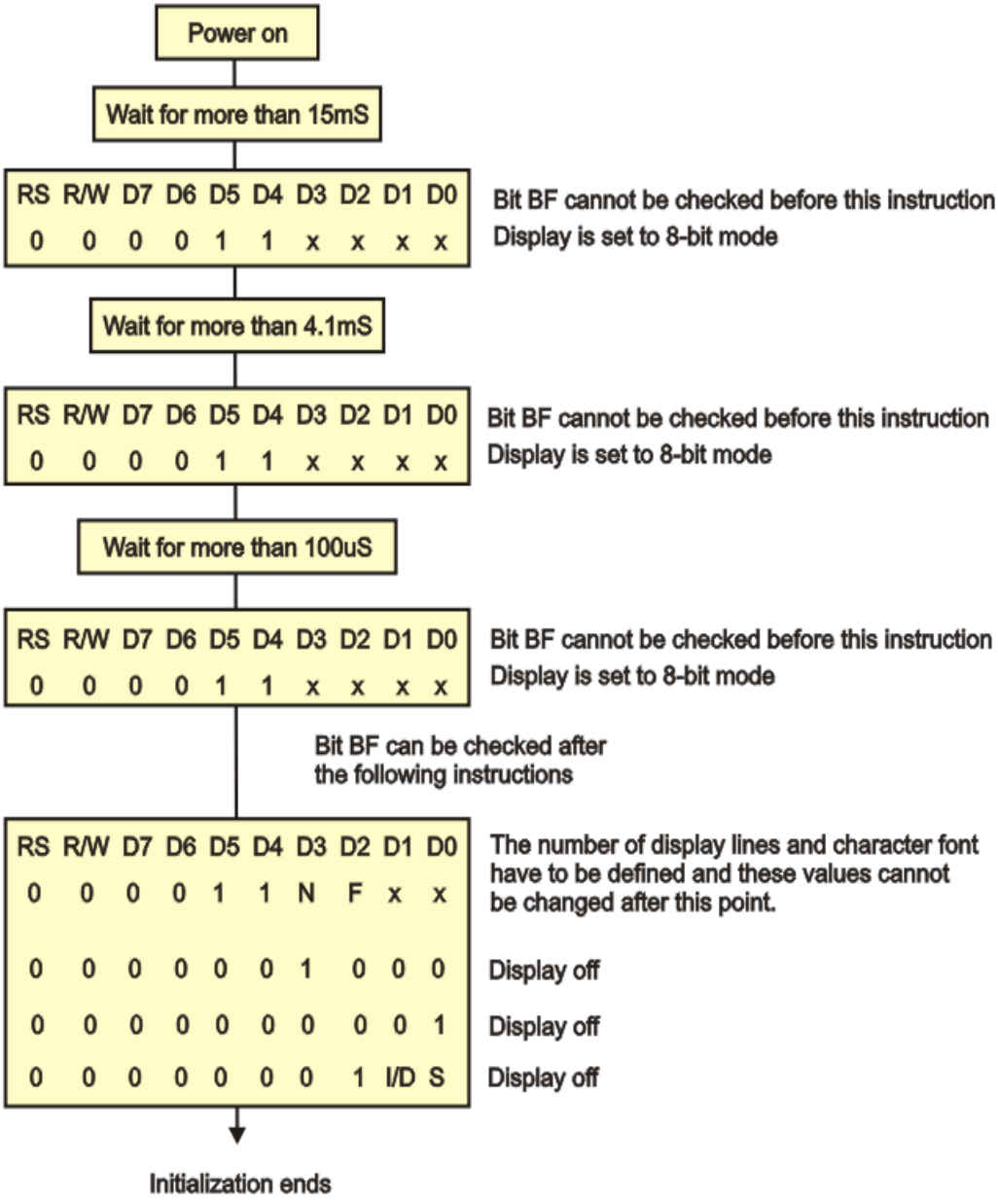

Figura 9. Algoritmo de inicialização no modo 8 bits.

Isto não é um erro!! Neste algoritmo, o mesmo valor é transferido três vezes em uma linha.

A figura 10 mostra o algoritmo de inicialização no modo de 4 bits.

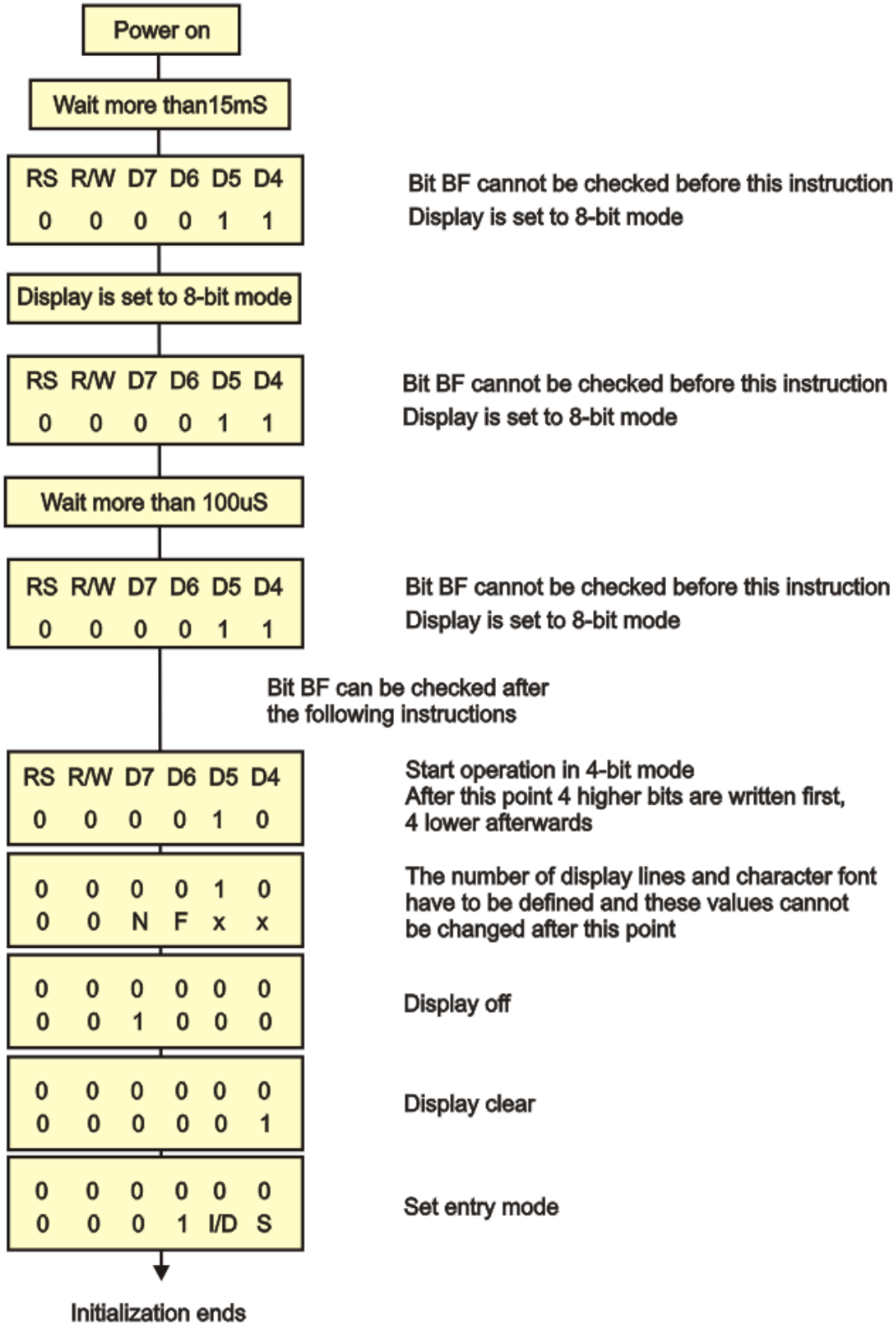

Figura 10. Algoritmo de inicialização no modo 4 bits.